TASTE OVER TIME: THE TEMPORAL DYNAMICS OF USER PREFERENCES

Joshua L. Moore, Shuo Chen, Thorsten Joachims
Cornell University, Dept. of Computer Science
{jlmo|shuochen|tj}@cs.cornell.edu

Douglas Turnbull
Ithaca College, Dept. of Computer Science
dturnbull@ithaca.edu

ABSTRACT

We develop temporal embedding models for exploring how listening preferences of a population develop over time. In particular, we propose time-dynamic probabilistic embedding models that incorporate users and songs in a joint Euclidian space in which they gradually change position over time. Using large-scale Scrobbler data from Last.fm spanning a period of 8 years, our models generate trajectories of how user tastes changed over time, how artists developed, and how songs move in the embedded space. This ability to visualize and quantify listening preferences of a large population of people over a multi-year time period provides exciting opportunities for data-driven exploration of musicological trends and patterns.

1. INTRODUCTION

Embedding methods are a class of models that learn positions for discrete objects in a metric space. Such models are widely employed in a variety of fields, including natural language processing and music information retrieval (MIR). In MIR, these methods find use in recommendation and playlist prediction, among other problems. Embedding methods offer advantages in two main aspects. First, they are often very easy to interpret: the resulting space can be easily visualized and inspected in order to explain the behavior of the model. Second, they can be applied to discrete objects without features, learning feature representations of the objects as part of model training.

In this work, we explore the use of embedding methods as a tool for identifying trends and patterns in multi-year listening data of Last.fm users. In particular, we propose novel time-dynamic embedding models that generate trajectories of musical preferences by jointly embedding listeners and the songs they play in a single metric space. In order to do this, we extend existing probabilistic playlists models [1, 2] by adding time dynamics, allowing users and songs to change position on a multi-year scale. By examining these models, we can draw conclusions about the behavior of listeners and musical artists over time. Based on these findings, we conjecture that these time-dynamic embedding methods provide exciting opportunities for data-driven exploration of musicological trends and patterns. To facilitate such research, scalable software implementations of our embedding methods are available at http://lme.joachims.org.

2. RELATED WORK

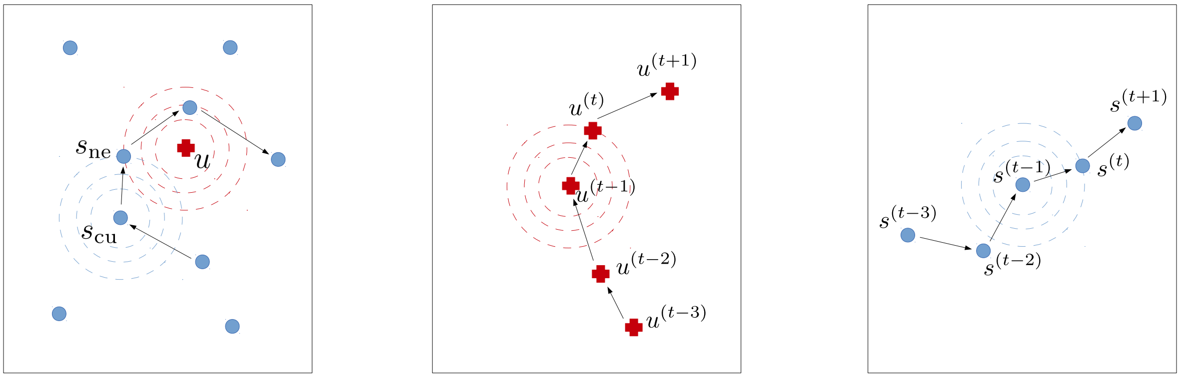
Embedding music into a low-dimensional space is useful for visualization and automatic playlist generation. There are numerous existing algorithms, such as Multi-Dimensional Scaling [7] and Local Linear Embedding [4], which have been applied to large corpora of songs and artists.

Our work is motivated by recent work of Moore et al. [2, 5] and Aizenberg et al. [1] on using historical playlist data, but we focus on long-term temporal dynamics. This is different from the short-term dynamics considered by Aizenberg et al., namely, the time of day of a track play by a station. This time dependency is employed to factor out the influence of different format blocks at a radio station based on the time of day (i.e. a college station may play classical music from 6 AM to 9 AM and jazz from 9 AM to noon). In our work, we allow the positions of users or songs to vary smoothly over the long-term, learning a representation for each three-month time step from the beginning of 2005 to the end of 2012. Second, both of these related works focus on automatic playlist prediction. In this paper, we instead use our model as a data analysis tool to explore long-range trends in the behavior of users, songs, and artists.

Weston et al. [9] use music embedding for a variety of MIR tasks including tag prediction and determining song similarity. Their embedding algorithm works by optimizing a rank-based loss function (e.g., AUC, precision at k) over training data for a given task. Our work differs from this in that our embedding results from a probabilistic sequence model that is learned from the track histories of users. In addition, the work by Weston et al. does not attempt to model the temporal dynamics.

Dror et al. [3] explore the use of temporal dynamics in collaborative filtering for music. However, the use of time dynamics in their work is mainly restricted to modeling biases for songs and users, which does not permit the visualization and analysis applications enabled by our work.

Finally, Shalit et al. [8] applies a dynamic topic model to audio features of songs for the purpose of modeling and



(a) (b) (c)

Figure 1: Illustrations of the embedding models. Blue dots and red crosses represent songs and users respectively. (a) Static playlist model. A playlist is represented by songs that are linked by arrows. The next song s_{ne} is decided by both current song s_{cu} and the user u (The popularity term also has its effect, which is not shown here). (b) The drifting of a user u over timesteps in the user-dynamic model. At each timestep, a random walk governed by a Gaussian distribution is taken. (c) Similar drifting of a song s over timesteps in the song-dynamic model.

discovering musical influence. While this work does not explicitly involve embedding songs, users or artists, it is a good example of the use of temporal dynamics in analysis of music data. In addition, their topic model requires audio features to represent each song. This is in contrast to our model where features are not required.

3. MODEL

In this section we detail the probabilistic embedding models we propose for temporal embedding of users and songs. Starting from a static playlist model (Section 3.1) similar to [2, 5], we incorporate a macroscopic temporal model under which the embedding can change over time, providing trajectories for users and songs through embedding space. In particular, we propose a user-dynamic embedding model (Section 3.2) in which users move against over a map of songs, as well as a song-dynamic embedding model (Section 3.3) in which songs move against a map of users. For both models, we briefly outline how they can be trained using maximum likelihood (Section 3.4).

3.1 Embedding Songs and Users for Playlist Modeling

Given a collection of songs $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and a collection of users $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$, each user's listening history can be described as a sequence of songs $p = (p^{[1]}, ..., p^{[k_p]})$ of length k_p, where each $p^{[i]} \in \mathcal{S}$. We refer to this (multi-year) sequence as the "playlist" of that user. The collection D of all user playlists is the training data for our embedding method.

Following the approach proposed in [2], we model a user's playlist using a first-order Markov model, but also augment it with a user model similar to [1]. As a result, the probability $\Pr(s_{\text{ne}}|s_{\text{cu}}, u)$ of the next song in a playlist depends only on the current song and the user. The overall probability of a playlist is therefore

$$\Pr(p|u) = \prod_{i=1}^{k_p} \Pr(p^{[i]}|p^{[i-1]}, u). \quad (1)$$

Note that transition triples $(s_{\text{ne}}|s_{\text{cu}}, u)$ (i.e., reads as "user u listened to the current song s_{cu}, then listened to the next song s_{ne}") are a sufficient statistic for this model.

Our goal is to embed each song and user into a d-dimensional latent Euclidean space such that song-song and song-user distances model the transition probabilities $\Pr(s_{\text{ne}}|s_{\text{cu}}, u)$. This provides such distances with a clear semantic meaning. More specifically, we want to learn a mapping $X(\cdot)$ that maps every song s or user u into that space, namely $X(s), X(u) \in \mathbb{R}^d$. The dimensionality d is manually specified. Alternatively, X can be considered as a $(|\mathcal{S}| + |\mathcal{U}|) \times d$ embedding matrix, the rows of which correspond to the position of songs and users in the latent space. We will not distinguish the two interpretations of X in the rest of the paper if it is clear from the context.

The specific model we propose for relating distances to transition probabilities is

$$\Pr(s_{\text{ne}}|s_{\text{cu}}, u) = \frac{e^{-\Delta(s_{\text{ne}}, s_{\text{cu}})^2 - \Delta(s_{\text{ne}}, u)^2 + b_{\text{idx}(s_{\text{ne}})}}}{Z(s_{\text{cu}}, u)}, \quad (2)$$

where $\Delta(x, y) = ||X(x) - X(y)||$ is the Euclidean distance between two embedded items (either song or user) in the latent space. $b_{\text{idx}(s)}$ is a scalar bias term that is added to model the popularity of each song, where $\text{idx}(s)$ returns the index for song s. For example, $\text{idx}(s_i) = i$. $Z(s_{\text{cu}}, u)$ is the partition function that normalizes the distribution. It is defined as

$$Z(s_{\text{cu}}, u) = \sum_{i=1}^{|\mathcal{S}|} e^{-\Delta(s_i, s_{\text{cu}})^2 - \Delta(s_i, u)^2 + b_i}. \quad (3)$$

Panel (a) of Figure 1 illustrates how song and user positions in the embedding space relate to the transition probability $\Pr(s_{\text{ne}}|s_{\text{cu}}, u)$. The red cross is the position of the user, and the blue dot labeled s_{cu} is the current song in the playlist. The probability of playing some song s_{ne} next depends on its sum of the squared distances to the current song and the user, plus its inherent popularity $b_{s_{\text{ne}}}$. This means that the transition to the next song s_{ne} is more likely if (1) the next song is close to the current song in the latent space, (2) the next song is close to the user('s taste) in the latent space, or (3) the next song is popular. We focus our experiments on two-dimensional embeddings, since this provides us with an X that can easily be visualized. However, higher-dimensional embeddings are possible as well.

3.2 User-dynamic Embedding Model

Combining Equations (1) and (2) models a playlist as a stochastic process on a microscopic level (i.e., on the timescale of minutes). In addition, we also model changes in user preferences as a stochastic process on a macroscopic level. In the following experiments, each macroscopic timestep $t \in \mathcal{T}$ ($\mathcal{T}$ is the set of all timesteps) denotes a quarter of a year, and notation like 20083 denotes "third quarter of year 2008".

Let us first consider a macroscopic stochastic process where positions of users are changing over time, while the position of the songs are fixed in the latent space. Denoting with $u^{(t)}$ the position of user u in embedding space at timestep t, the overall trajectory of a user is $u^{(*)} = (u^{(1)}, u^{(2)}, ...)$. At each timestep t, the microscopic transition probability $\Pr(s_{\text{ne}}|s_{\text{cu}}, u^{(t)})$ now depends on the users current position, and the conditional probability of the next song is

$$\Pr(s_{\text{ne}}|s_{\text{cu}}, u^{(t)}) = \frac{e^{-\Delta(s_{\text{ne}},s_{\text{cu}})^2 - \Delta(s_{\text{ne}},u^{(t)})^2 + b^{(t)}_{\text{idx}(s_{\text{ne}})}}}{Z(s_{\text{cu}}, u^{(t)})}. \quad (4)$$

Note that even though the positions of songs are fixed, we still give each song a time-varying popularity term $b_i^{(t)}$.

To restrict users from drifting too much from one timestep to the other, we model a users trajectory as a Gaussian random walk. Panel (b) of Figure 1 illustrates such a random walk. Concretely, this means that the user's next position $u^{(t)}$ is a Gaussian step $\mathcal{N}(u_i^{(t-1)}, \frac{1}{2\nu_{\text{user}}} I_d)$ from the current position $u^{(t-1)}$. Here, I_d is the d-dimensional identity matrix, and ν_{user} is the variance (which can be viewed as a regularization coefficient that influences step sizes). This Gaussian distribution makes it more likely that the user's positions at two consecutive timesteps are close to each other.

Considering both the stochastic process over transition triples and the stochastic process describing the users' trajectories, the overall user-dynamic embedding model can be trained via maximum likelihood. The resulting optimization problem is

$$\max_{\substack{X \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{T}||\mathcal{U}|)\times d} \\ b \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{S}|}}} \prod_{(s_{\text{ne}}|s_{\text{cu}},u^{(t)})\in D} \Pr(s_{\text{ne}}|s_{\text{cu}}, u^{(t)}) \cdot \prod_{i=1}^{|\mathcal{U}|} \prod_{\substack{t\in\{\tau|(\tau\in\mathcal{T}) \\ \wedge(\tau-1\in\mathcal{T})\}}} e^{-\nu_{\text{user}}\Delta(u_i^{(t-1)},u_i^{(t)})^2}, \quad (5)$$

where the song and time-dependent user positions are optimized to maximize the likelihood of the observed playlists.

3.3 Song-dynamic Embedding Model

Similar to the user-dynamic embedding model, we also consider a song-dynamic embedding model which fixes the position of users and allows songs to drift over time. In this model, the probability of each transition triple is

$$\Pr(s^{(t)}_{\text{ne}}|s^{(t)}_{\text{cu}}, u) = \frac{e^{-\Delta(s^{(t)}_{\text{ne}},s^{(t)}_{\text{cu}})^2 - \Delta(s^{(t)}_{\text{ne}},u)^2 + b^{(t)}_{\text{idx}(s_{\text{ne}})}}}{Z(s^{(t)}_{\text{cu}}, u)}. \quad (6)$$

After introducing an analogous Gaussian random walk for songs over different timesteps (as illustrated in Panel (c) of Figure 1), we get the training problem

$$\max_{\substack{X \in \mathbb{R}^{(|\mathcal{T}||\mathcal{S}|+|\mathcal{U}|)\times d} \\ b \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{S}|}}} \prod_{(s^{(t)}_{\text{ne}}|s^{(t)}_{\text{cu}},u)\in D} \Pr(s^{(t)}_{\text{ne}}|s^{(t)}_{\text{cu}}, u) \cdot \prod_{i=1}^{|\mathcal{S}|} \prod_{\substack{t\in\{\tau|(\tau\in\mathcal{T}) \\ \wedge(\tau-1\in\mathcal{T})\}}} e^{-\nu_{\text{song}}\Delta(s_i^{(t-1)},s_i^{(t)})^2}, \quad (7)$$

where users and time-dependent song positions are optimized.

From a technical perspective, it is conceivable to train an embedding model with both users and songs varying their position over time, which will output an embedding matrix X of $(|\mathcal{T}|(|\mathcal{S}| + |\mathcal{U}|))$ rows. We briefly explored this model, but found it difficult to interpret the resulting trajectories. We therefore focus on the restricted models in our empirical evaluation.

3.4 Training of Probabilistic Embedding Models

The maximum likelihood optimization problems in Equations (7) and (5) are of substantial scale. Previous sequence models were trained using stochastic gradient methods [1, 2, 5]. However, those training algorithm does not scale well, since the complexity of each training iteration is quadratic in the number of terms in the partition function (in our case $|\mathcal{S}|$). In related work on (non-temporal) sequence modeling for natural language [6], we developed an approximate, sampling-based training algorithm which estimates the partition function on the fly. This training procedure has complexity which is only linear in the number of terms in the partition function, and we adopt this algorithm for training. A software package implementing the training algorithm is available online at `http://lme.joachims.org`.

4. EXPERIMENTS

Our experiments revolve around a *Last.fm* data set which we crawled using the site's API [1]. The crawl was conducted over the course of several weeks in the fourth quarter of 2012. Although it is unused in this work, we were initially also interested in the social network data, so we

[1] http://www.last.fm/api

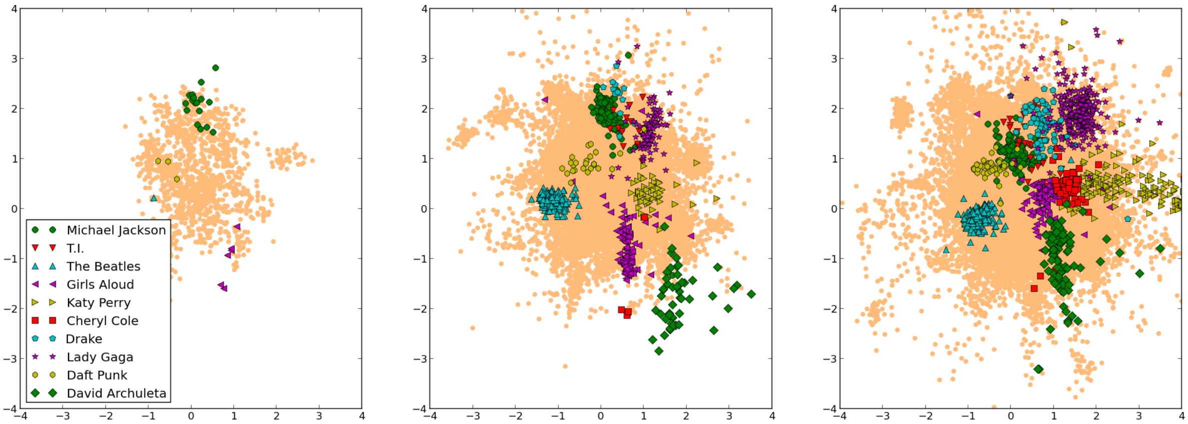


Figure 2: The song-dynamic model's song space plotted (from left to right) at 20051, 20091, and 20124

crawled through the social network using the top listener for each of the 10 top artists on the site at the time as seeds. For each user, we crawled the user's complete timestamped track history and friends list. We later augmented this data with the age, gender, and country of each user (for those for which it was available). We also crawled the tags for some of the songs, although we do not take advantage of this data in this work.

The result contains over 300,000,000 track plays by roughly 4700 users, with over 550,000 unique tracks. This data contains many noisy track names, so we pruned the data further by only considering tracks with at least 1000 plays and discarding users with no remaining track history after infrequent songs are discarded. This yields the set of track histories used in the experiments, which contains 4,551 users, 32,401 unique tracks, and roughly 200,000,000 track plays. We used this to create our "per-user playlist" data by splitting the track histories into playlists of consecutive songs that were played within 15 minutes of each other. Finally, we quantized the timestamps to divide each user's track history into year quarters, ranging from first quarter, 2005 until fourth quarter, 2012, for a total of 32 timesteps. From this point on, we will refer to the nth quarter of year $yyyy$ as $yyyyn$, such as 20051 for 2005 first quarter.

We considered models with 2 dimensions in this work for the sake of simplicity and ease of visualization. In order to find good values for ν_{song} and ν_{user}, we further divided the data by placing each fifth song transition into a validation set and the rest into the training set. We then used these to validate for the optimal values of these parameters. The user-dynamic model performed best with a low value of ν_{user}, with its optimal value at 0.01. In contrast, the song-dynamic model performed best with strong regularization, and the optimal ν_{song} was found to be 2.0.

4.1 Demographics of users

The demographics of the data set reflect characteristics of the average Last.fm user. For each demographic category, we report percentages based on the number of users reporting in that category. 83% reported an age, 89% reported a country, and 91% reported gender. In our data, about 78% of the users are male, and about 88% are between the ages of 15 and 25 (roughly evenly split between the two groups) as of the crawl in 20124. The median user age is 20, and the average is about 20.8. Due to the social network crawl and a coincidence of the seed users, roughly 57% of our users are from Brazil. The country distribution has a fairly long tail, with only 84% coming from the 10 most popular countries, and 91% coming from the 20 most popular countries. The ten most well-represented countries in the data set are Brazil (57%), US (8%), UK (4%), Poland (3%), Russia (2.6%), Germany (2.3%), Spain (1.6%), Mexico (1.6%), Chile (1.3%), and Turkey (1.1%).

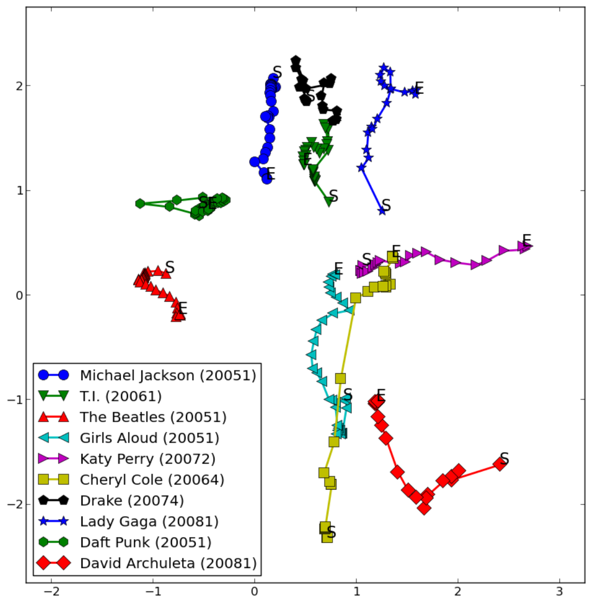


Figure 3: Artist trajectories over time. The legend gives the first quarter in which each artist was observed

4.2 Song-dynamic Model

In the song-dynamic model, songs can move over time through a map of users. Among other things, the resulting trajectories give insight into how the appeal of songs and artists changed over time.

In Figure 2, we show the embedding of the songs at the start, middle, and end of the time sequence (i.e., timesteps

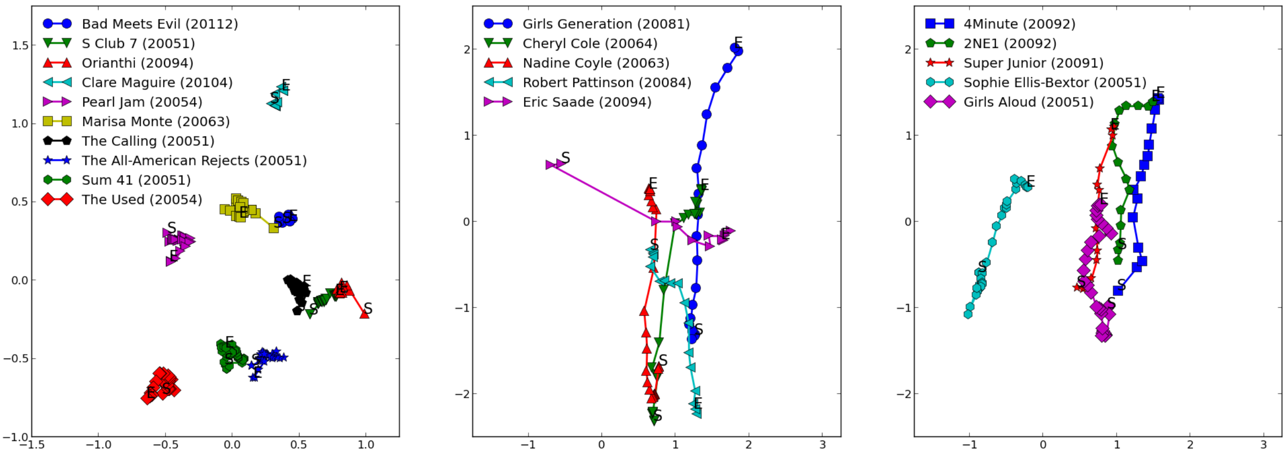


Figure 4: The 10 artists with the smallest variance in position over time (left) and the 10 with the largest variance in position over time (top 5 in center, next five at right). The first timestep at which each artist was observed is listed in parentheses.

20051, 20091, and 20124). A song is plotted once it has been played at least once, which explains why the space becomes more dense over time. The locations of users are not plotted to reduce clutter. Generally speaking, the density of users is greatest around the origin and then decreases outwards. In this sense more popular music lies in the center, but note that we also capture popularity through the specific song bias parameter.

Are similar songs embedded at similar locations? To illustrate the semantic layout of the embedding space, we highlight the songs of some reference artists. Note that the songs of the reference artist cluster even though our embedding method has no direct information about artists. This verifies that the model can indeed learn about songs similarity merely from the listening patterns. We also note that our intuitive notion of artist similarity generally matches the distance at which our model positions them in embedding space.

How do songs and artists move? Figure 2 also shows that the songs of some artists move in the embedding space, while others remain more stationary. The artists' changes are aggregated into trajectories and displayed in Figure 3. Each dot in Figure 3 indicates the mean location of the songs of one artist at a specific time step. This plot enables us to see more clearly some events and trends in the music world that influence the model.

First, note that Michael Jackson's trajectory starts off clumped together in the same space, moving very little. Then, after some number of timesteps, it starts moving quickly towards the center. Upon closer inspection, the turning point in this trajectory turns out to line up exactly with the death of Michael Jackson in June, 2009.

Similarly, the Beatles start to drift slightly away from the center as many other artists enter the model. Then, they make an abrupt turn back towards the center. This aligns with the release of the Beatles' full catalog on iTunes in the 20104 after being totally unavailable via digital distribution before then.

Daft Punk also starts to drift away from the center until the release in December, 2010 of the motion picture *Tron: Legacy*, which featured a popular soundtrack by the duo.

We can also see Girls Aloud and Cheryl Cole (of Girls Aloud) drift from the edges rapidly towards the center in correlated paths, and the emergence of David Archuleta, an American Idol runner-up in May, 2008. All follow a similar trajectory in user space, indicating that the users that previously listened to Girls Aloud are listening to David Archuleta a few years later.

We can also see artists like Katy Perry and Lady Gaga drift away from the center after the peak of their popularity, and we see Drake drift towards the center in what can partly be explained by a shift in his style from something more hip hop oriented to a somewhat more poppy style.

What does the variance of a trajectory indicate? The trajectories are useful not only for visualization, but also as the basis for further aggregating and quantifying the behavior of an artist. Figure 4 shows the artists with the smallest and largest variance in position over time. The specific criterion used here for a given artist is the average distance over timesteps from the artist's embedding at that timestep to the mean vector of that artist's representation over all time steps. To avoid obscure artists that would be difficult to interpret without further background knowledge, we only consider artists who appeared in the track histories of at least 10% of the users.

The left-hand panel in Figure 4 shows the 10 artists with small variance. Many of these are well-established artists that probably undergo little change in style or fan base.

The panel in the middle and on the right-hand side of Figure 4 show the 10 artists with the largest variance. Many of these are popular artists that have a large change in appeal – i.e., those that go from being relatively obscure to quite popular.

The variance of a trajectory in only one possible statistic that summarizes a path. We conjecture that other summary statistics will highlight other aspects of an artist's development, providing additional criteria for exploratory data

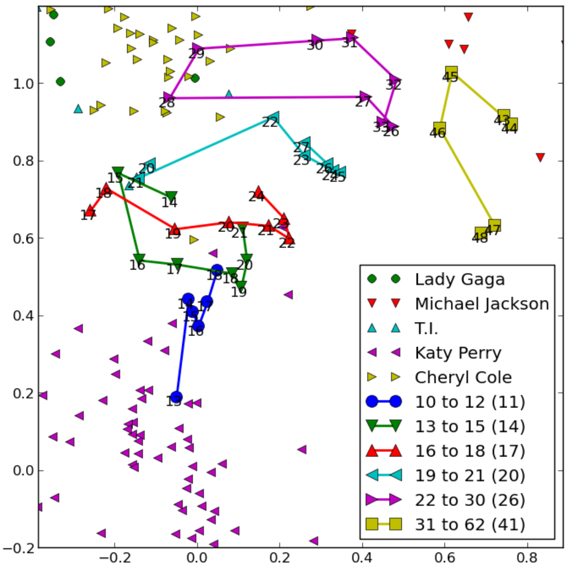


Figure 5: Trajectories of users with age, grouped by age in 2005. Each point is labeled with the average age of the group at that time. The legend also gives the average age in 2005 of the users in that group (in parentheses).

analysis.

4.3 User-dynamic model

The user-dynamic model is dual to the song-dynamic model, in that it models trajectories of users on a map of songs. While the trajectories of indiviual users provide an intersting tool for reflection, they are difficult to interpret for outsiders. We therefore only show aggregate user paths.

One such aggregation is shown in Figure 5. Here, we can see the behavior of users when aggregated by age. Specifically, the users are grouped by age in 2005 in order to separate the effect of a person's absolute age from the effect of the change in the average listener's taste profile.

Distinctive differences in trajectory can be seen, with the youngest group moving to north, away from Katy Perry and many other more "sugary" pop artists, and towards more dance and R&B oriented pop artists as well as the hip hop cluster which is further north, outside the figure.

The other age groups see more lateral moves and tend to be further north, even when age is fixed. The oldest age groups (where 22 to 30 and 31 to 62 were aggregated with a larger interval due to a smaller number of users in these age ranges) start very far north, and the 31 to 62 group mostly hovers around the eastern part of the figure. Outside of the figure and to the right are where many older rock bands such as the Rolling Stones and the Beatles lie, and this oldest age group is also closer to them.

5. CONCLUSIONS

We presented novel probabilistic embedding methods for modeling long-term temporal dynamics of sequence data. These models jointly embed users and songs into a metric space, even when no features are available for either one. Users and/or songs are allowed to change position over time, which enables the analysis of long-term dynamics of user tastes and artist appeal and style. The ability to visualize the learned embeddings is a key feature for easy interpretability and open-ended exploratory data analysis. We conjecture that such embedding models will provide interesting tools for analyzing the growing body of listening data. Furthermore, the embedding models described in the paper can easily be adapted and extended to include further information (e.g., social network data), providing many directions for future work.

5.1 Acknowledgements

This work was supported by NSF grants IIS-1217485, IIS-1217686, and IIS-1247696. The first author is supported by an NSF Graduate Research Fellowship. We would also like to thank the anonymous reviewers for their feedback, and Brian McFee for helpful discussions and technical advice.

6. REFERENCES

[1] N. Aizenberg, Y. Koren, and O. Somekh. Build your own music recommender by modeling internet radio streams. In *Proceedings of the 21st international conference on World Wide Web*, pages 1–10. ACM, 2012.

[2] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 714–722. ACM, 2012.

[3] G. Dror, N. Koenigstein, and Y. Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.

[4] V. Jain and L. Saul. Exploratory analysis and visualization of speech and music by locally linear embedding. *ICASSP*, 2004.

[5] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull. Learning to embed songs and tags for playlist prediction, 2012.

[6] J. L. Moore and T. Joachims. Fast training of probabilistic sequence embedding models with long-range dependencies. *Arxiv pre-print*, 2013.

[7] J. Platt. Fast embedding of sparse music similarity graphs. *NIPS*, 2004.

[8] U. Shalit, D. Weinshall, and G. Chechik. Modeling musical influence with topic models. In *ICML*, 2013.

[9] J. Weston, S. Bengio, and P. Hamel. Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval. *Journal of New Music Research*, 40(4):337–348, 2011.